

¿Como resolver?

Johann creó una excepción `SaldoInsuficienteException`, como se muestra en el último video:

```
public class SaldoInsuficienteException extends Exceptio

    public SaldoInsuficienteException(String msg) {
        super(msg);
    }
}
```

[COPIA EL CÓDIGO](#)

Luego cambió la implementación del método *sacar* para lanzar una excepción:

```
public abstract class Cuenta {

    //atributos y otros métodos omitidos

    public void sacar(double valor) throws SaldoInsufici
        if(this.saldo < valor) {
            throw new SaldoInsuficienteException("Saldo:
        }
        this.saldo -= valor;
    }
}
```

[COPIA EL CÓDIGO](#)

Hasta ahora todo está bien, pero la implementación del método *sacar* se basa en la clase *CuentaCorriente* que no quiere compilar:

```
public class CuentaCorriente extends Cuenta implements T
//atributos y otros métodos omitidos

@Override
public void sacar(double valor) {
    double valorASacar = valor + 0.2;
    super.sacar(valorASacar);
}
}
```

[COPIA EL CÓDIGO](#)

¿Cuál es el problema?



Como la excepción es un *checked*, debemos usar *throws* **SaldoInsuficienteException* en la firma del método *sacar* de la clase *CuentaCorriente*.



Correcto. Observe que llamamos a *super.sacar(..)* en el método *sacar* de la clase *CuentaCorriente*. El compilador se da cuenta de que el método de la clase madre tiene el *throws* de una excepción *checked* y obliga al método hijo una acción.

B

Como la excepción es un *checked*, debemos eliminar *SaldoInsuficienteException* en la firma del método *sacar* de la clase *Cuenta*.



C

Como la excepción es *unchecked*, no es necesario utilizar *throws SaldoInsuficienteException* en la firma del método de *sacar* de la clase *CuentaCorriente*.

**PRÓXIMA ACTIVIDAD**